Forced Sequence Sequential Decoding

A concatenated coding system with iterated sequential inner decoding

Jensen, Ole Riis; Paaske, Erik

Published in:
I E E E Transactions on Communications

Link to article, DOI:
10.1109/26.725306

Publication date:
1998

Document Version
Publisher's PDF, also known as Version of record

Forced Sequence Sequential Decoding: A Concatenated Coding System with Iterated Sequential Inner Decoding

Ole Riis Jensen and Erik Paaske

***Abstract*— In this paper, we describe a new concatenated decoding scheme based on iterations between an inner sequentially decoded convolutional code of rate $R=1/4$ and memory $M=23$, and block interleaved outer Reed–Solomon (RS) codes with nonuniform profile. With this scheme decoding with good performance is possible as low as $E_b/N_0=0.6$ dB, which is about 1.25 dB below the signal-to-noise ratio (SNR) that marks the cutoff rate for the full system. Accounting for about 0.45 dB due to the outer codes, sequential decoding takes place at about 1.7 dB below the SNR cutoff rate for the convolutional code. This is possible since the iteration process provides the sequential decoders with side information that allows a smaller average load and minimizes the probability of computational overflow. Analytical results for the probability that the first RS word is decoded after C computations are presented. These results are supported by simulation results that are also extended to other parameters.**

***Index Terms*—Concatenated coding, convolutional codes, Fano decoding, forced sequence decoding, sequential decoding, terative decoding.**

I. Introductory Concepts

THE CODING system that we will present in this paper is an iterative, serially concatenated coding system, which is weakly based on the recommendations of [1] and the philosophy of iterative systems, which was developed independently by Collins and Hizlan in [2] and by Paaske in [3]. However, we believe that the bootstrap technique, which is inherent in any iterative decoding system was first explored by Jelinik and Cocke in [4].

The system we shall consider consists of outer Reed–Solomon (RS) codes, a block interleaver, and sequentially decoded inner convolutional codes, as shown in Fig. 1. Because of the *iterative* nature of the interaction between the inner and outer decoders, we denote this scheme *a concatenated coding system with iterative forced sequence sequential inner decoding* (FSSD).

As in many other systems we will transmit the information sequence in blocks called *transfer frames* of fixed length as shown in Fig. 2. The outer code encodes the information sequence in blocks and fills the frame with codewords by columns, whereupon a *block interleaver* rearranges the frame and inputs the frame data by rows to the inner encoder.

The encoded information sequence is transmitted on the channel using antipodal signaling (e.g., BPSK) with energy E_s per transmitted symbol. Assuming that the channel adds additive white Gaussian noise (AWGN) with zero-mean and power spectral density $N_0/2$, the received signal will be

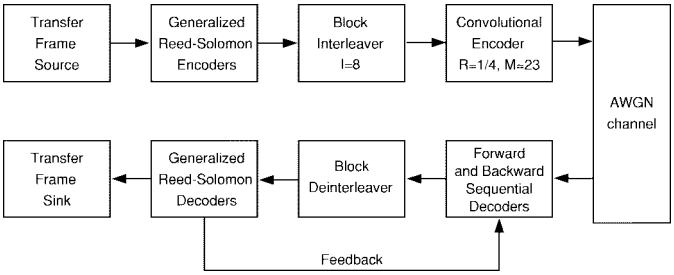

Fig. 1. Concatenated coding system with iterated sequential inner decoding.

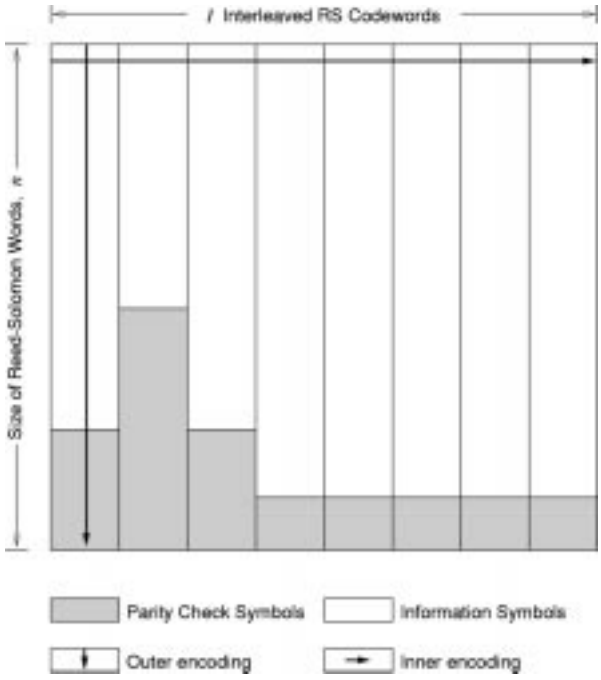

Fig. 2. Frame structure for the concatenated FSSD scheme.

Paper approved by S. B. Wicker, the Editor for Coding Theory and Techniques of the IEEE Communications Society. Manuscript received May 1, 1997; revised March 30, 1998. This paper was presented in part at the 1995 IEEE International Symposium on Information Theory, Whistler, BC, Canada, September 17–22, 1995.

The authors are with the Department of Telecommunication, Technical University of Denmark, DK-2800 Lyngby, Denmark (e-mail: riis@tele.dtu.dk; ep@tele.dtu.dk).

Publisher Item Identifier S 0090-6778(98)07767-8.

normally distributed $\mathcal{N}(\pm\sqrt{E_s},\ \sqrt{N_0/2})$. The modem demodulator performs soft decisions, but instead of using $Q = \infty$, the received signal is quantized into $Q = 16$ levels to facilitate metric calculations in our simulations. The crossover probabilities for the resulting binary input, Q-ary output (BIQO), channel depend on the settings of the $Q-1$ channel quantizer thresholds.

For a Viterbi decoder it is common practice to let the thresholds vary linearly with the standard deviation of the noise, $\sigma = \sqrt{N_0/2}$, since very good results are achieved this way. In our work we have followed this practice.

Systems that transmit in frames often use an attached synchronization marker placed in the beginning of the frame, which is normally not encoded by the outer encoder, but in most cases by the inner encoder. It is known that the synchronization problem can be solved efficiently [3], with a rather short synchronization marker, and we shall assume perfect synchronization, henceforth.

The outer decoding system is based on Reed–Solomon codes with symbol size $J = 8$ and codeword block length $n = 255$. Given an error correcting capability of T errors, the dimension of the code becomes $k = n - 2T$, resulting in $n - k = 2T$ parity check symbols per codeword and a minimum distance of $d_{\min} = 2T + 1$.

Since the RS codes are symbol oriented (in opposition to the inner code which is bit oriented), they are very efficient for decoding errors that appear in bursts such as those produced by the inner decoder. This is our main reason for using RS codes as the outer code in the FSSD scheme, and as such RS codes have found wide use in space applications.

If t is the number of errors and e is the number of erasures in a received word, correct decoding is performed if

$$2t + e \le n - k = d_{\min} - 1. \tag{1}$$

If (1) is not satisfied one of two situations will occur, either the decoder will detect that the word cannot be corrected and declare a *decoding failure*, or it will decode the word into a wrong codeword, i.e., make a *decoding error*. These two situations happen with probabilities P_F and P_E, respectively, and we shall exploit the error detection ability when in Section II we define how the FSSD iteration scheme works.

The probability, P_E, that a received word with $2t + e > d_{\min} - 1$ errors and erasures is decoded into a wrong word is upper bounded [5] by

$$P_E \le 1/(T - \lceil e/2 \rceil)!. \tag{2}$$

Although (2) is an upper bound it is rather tight and therefore it may serve as a useful approximation. Since the error detection capability of the RS decoder is only useful when P_E is small compared to P_F, it is obvious that the number of erasures cannot be increased beyond a certain level, and we shall define a maximum allowable number of erasures for each RS word. We denote the maximum error correction capability of the ith RS word by T_i, $1 \le i \le I$, when there are no erasures present, and by $T_{\min,i}$ when the maximum allowable number of erasures is present. Thus, the maximum number of allowable erasures in the ith RS word becomes $e_i = 2(T_i - T_{\min,i})$.

We shall assume an interleaving degree of $I = 8$ and the error and erasure correcting capabilities of I interleaved RS codewords of the same length, n, constitute the *profile* of the outer RS codes. The profile is written as

$$\mathrm{RS}_I = [T_1, T_2, \cdots, T_I],\ [T_{\min,1}, T_{\min,2}, \cdots, T_{\min,I}].$$

If $T_i = T_{\min,i}$ for all i, the second vector may be omitted. If $T_1 = T_2 = \cdots = T_I$ and $T_{\min,1} = T_{\min,2} = \cdots = T_{\min,I}$ the profile is said to be *uniform*, otherwise the profile is *nonuniform*, as shown in Fig. 2. If one or more of the words have no coding, i.e., if $T_i = 0$, the profile is said to be *sparse*. It will later become clear that the interleaving degree should not be chosen too small, and we note here that higher values could turn out to provide better results.

The inner code is an (N, M) convolutional code with rate $R = 1/N$, and memory M (or constraint length $K = M+1$), and it is specified by an encoder matrix

$$\mathbf{G}(D) = [G_1(D), G_2(D), \cdots, G_N(D)]$$

where $G_i(D)$ is the ith generator polynomial of degree M or less

$$G_i(D) = g_{0,i} + g_{1,i}D + \cdots + g_{M,i}D^M, \qquad 1 \le i \le N$$

where $g_{j,i}$ are binary digits, and D is the Huffman delay operator. Similarly, the source (input) sequence $\mathbf{x}$ of length $L+1$ is represented by the polynomial

$$x(D) = x_0 + x_1 D + \cdots + x_L D^L$$

and an encoding of the input sequence $x(D)$ is mathematically described as the product

$$\mathbf{y}(D) = x(D)\mathbf{G}(D).$$

Two parameters are often considered to be the most important ones of a convolutional code. These are the *free distance* and the *column distance function*, where the free distance, d_f, is defined as the minimum Hamming weight, W_{H}, of all codewords, except the all-zero codeword

$$d_f = \min_{x(D)\neq 0} \{W_{\mathrm{H}}(\mathbf{y}(D)) \,|\, \mathbf{y}(D) = x(D)\mathbf{G}(D)\}.$$

The ith component, d_i^c of the column distance function, $\mathbf{d}^c$, is defined as the minimum Hamming weight of all codewords having a nonzero first branch and truncated after $i+1$ branches. The first $M+1$ components of $\mathbf{d}^c$ is denoted the *distance profile*, DP, of the code. For two codes with the same rate and memory and distance profiles DP_1 and DP_2, we say that

$$\mathrm{DP}_1 \text{ is superior to } \mathrm{DP}_2 \text{ if } \begin{cases} d_i^{c_1} = d_i^{c_2}, & 0 \le i < k \\ d_k^{c_1} > d_k^{c_2}, & i = k \end{cases} \tag{3}$$

where k is the first position where the column distance functions differ. A code with a distance profile that is as good as the distance profile of any other code with the same rate and memory is called an optimum distance profile (ODP) code.

The number of paths searched by a sequential decoder is minimized when the convolutional code has an optimum distance profile, and since we intend to perform both forward and backward decoding, it is essential that the chosen code

has a good distance profile in both directions. Most of the ODP codes listed in the literature result in codes with very bad distance profiles when the encoders are reversed. A code search was therefore performed with a rate $R = 1/4$, memory $M = 23$, systematic ODP encoder matrix found in [6] as seed. This encoder matrix was multiplied by all binary polynomials of degree $M \leq 23$ and truncated to degree $M = 23$. All resulting encoders generate a code with the first $M + 1$ components of their column distance functions identical to the ones of the original code [7], [8]. The encoder chosen as the best one was the one that generates the code with the largest free distance *and* where the code generated by the reversed encoder matrix has the best distance profile. The chosen nonsystematic encoder matrix (in the usual octal notation) is

$$\mathbf{G} = [45523573,\ 55765147,\ 64121135,\ 70342331]$$

with free distance $d_f = 54$, and the distance profiles of the code and its reversed counterpart are

$$\begin{aligned}\mathrm{DP} = &[4,\ 6,\ 8,\ 9,\ 10,\ 11,\ 13,\ 14,\ 15,\ 17,\ 18,\ 19,\ 20,\ 20,\\ &21,\ 23,\ 24,\ 25,\ 25,\ 27,\ 27,\ 29,\ 30,\ 30]\\ \mathrm{DP}_{\mathrm{rev}} = &[4,\ 6,\ 8,\ 9,\ 10,\ 11,\ 11,\ 13,\ 14,\ 15,\ 16,\ 17,\ 18,\ 20,\\ &20,\ 21,\ 22,\ 23,\ 24,\ 24,\ 26,\ 26,\ 27,\ 27].\end{aligned}$$

Since the latter distance profile is slightly worse than the first (optimum) one, we can expect that the forward decoder will perform slightly better than the backward decoder.

The encoding process of convolutional codes may be illustrated as a code tree, and in fact convolutional codes are a linear subset of *tree codes*. The input to the encoder determines whether to go up or down one branch in the code tree, and values along each branch correspond to the output from the encoder. The branch and its successors stemming from the correct path at depth i is called *the ith incorrect subtree* and the number of visits to nodes in these subtrees determines the computational distribution of each bit along the correct path. We will discuss this distribution more thoroughly below since it is the basis of our analysis of the FSSD scheme.

II. The Iteration Scheme and Forced-Sequence Decoding

In the FSSD system that we propose here, the inner code is decoded by a sequential decoder. A number of different sequential decoding algorithms have been suggested, all with different benefits and drawbacks, and the two best known algorithms are the Stack and the Fano algorithms which are described in more detail in [9]–[11]. In this paper we shall report only on results obtained with the Fano algorithm. The Fano algorithm falls into the category of *metric first* sequential algorithms, which means that it always favors the path(s) that currently have the best metric among the explored paths so far. The Fano metric [11] is used to evaluate the paths.

The actual number of nodes explored by a sequential decoder is a random variable, C, i.e., a sum of random variables, $C = C_1 + C_2 + \cdots + C_L$, where C_i may be interpreted as the number of computations needed to decode the ith bit. For the Fano algorithm C_i is defined as the total number of visits to nodes in the ith incorrect subtree. The average number of computations per decoded bit, $C_{AV} = C/L$, is also a random variable, and both C_i and C_{AV} are essentially Pareto distributed for large C, i.e., we may assume that

$$P[C_i > C] \approx AC^{-\rho} \quad \text{and} \quad P[C_{AV} > C] \approx A_d C^{-\rho} \tag{4}$$

hold for large C and that asymptotically the Pareto distributions give accurate descriptions of the computational distributions. Here, A and A_d are relatively unimportant constants and the *Pareto exponent*, ρ, solves $R = E_0(\rho)/\rho$ where $E_0(\rho)$ is the Gallager function for the BIQO channel:

$$E_0(\rho) = 1 + \rho - \log_2 \sum_j \left(p(z_j|0)^{1/(1+\rho)} + p(z_j|1)^{1/(1+\rho)}\right)^{1+\rho}. \tag{5}$$

The Pareto distribution is a very unfortunate one because $P[C_{AV} > C]$ decreases less than exponentially with C, and this implies a large mean and a large variance, especially on a poor channel. The ith moment of the Pareto distribution is given by

$$E(C^i) = \lim_{C \to \infty} \frac{A\rho}{i - \rho} \left(C^{i-\rho} - 1\right) \tag{6}$$

and for the Pareto distribution to have a bounded mean value, we must require that $\rho > 1$, or equivalently $R < R_0 = E_0(1)$, where R_0 is denoted as the *computational cutoff rate*.

If L is the total frame length in bits after outer encoding, then $C_{\mathrm{lim}} L$ denotes the maximum number of computations that can be allowed for decoding of each frame. The probability of a *computational overflow* is thus

$$P_{OF} = P[C_{AV} > C_{\mathrm{lim}}] \approx A_d C_{\mathrm{lim}}^{-\rho}. \tag{7}$$

Frames that suffer a computational overflow are lost, or *erased*, and the probability of this unfortunate event increases rapidly with decreasing E_s/N_0. For $R > R_0$ we have $\rho < 1$, and the mean of the distribution is unbounded, which implies that decoding becomes more difficult, but not impossible. Depending on C_{lim} we will still be able to decode most of the frames, but some of the frames will become erasures, and this is in fact the major problem of all sequential decoding schemes. One way to beat this problem is to provide the inner decoders with some kind of *side information*. In the FSSD scheme, the side information is provided through the iteration process between the inner and outer decoders as described in more detail below.

The complexity of the sequential decoder is, however, almost independent of the constraint length of the convolutional code. Furthermore, by selecting a large M, the decoder error probability can be made arbitrarily small. Thus, by selecting the $M = 23$ code mentioned above, we may assume that frames that did not suffer an overflow have very small error probability.

If the channel is assumed to suffer a single very severe noise burst that causes a computational overflow, half the frame length will on the average be left undecoded. Thus, the average bit error probability of such frames will be 25%, and it is therefore unlikely that the outer RS decoding will

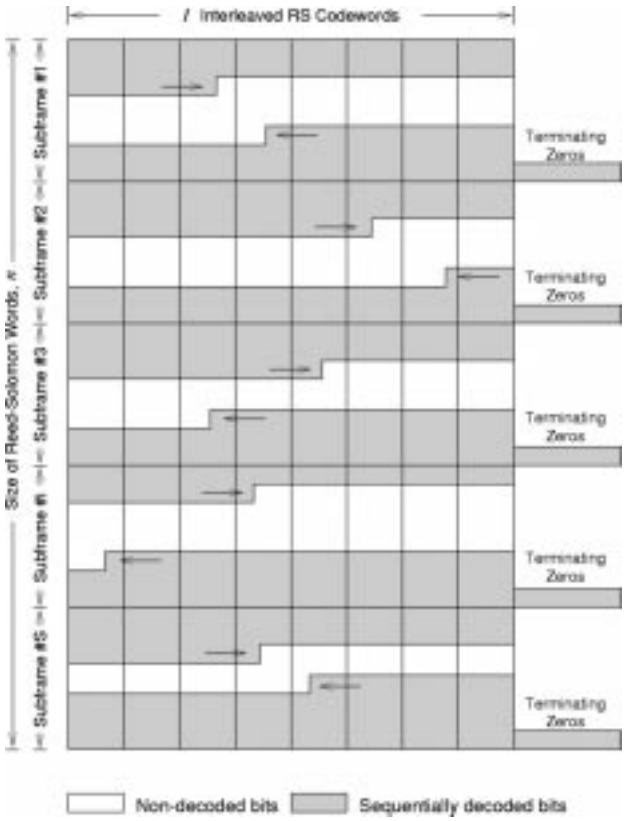

Fig. 3. Frame showing sequentially decoded and nondecoded parts. Arrows indicate decoding directions of the inner decoders.

be successful. To avoid this problem, which increases with decreasing SNR, we will use *bidirectional* decoding, i.e., both forward and backward decoding starting from each end of the frame. Again assuming only one severe noise burst in a particular frame, the two decoders will approach the burst from both ends. Even if the decoders are unable to penetrate the burst only a small part of the frame, approximately equal to the length of the noise burst, will be left undecoded, and consequently, successful outer decoding is much more likely. The probability that a frame contains more than one severe noise burst of course increases with longer frames, and to avoid this, we will split the full frame into S *subframes* which are decoded bidirectionally and independently in parallel. Thus, each subframe must be terminated separately with M zeros, and the length of each subframe becomes $L_{\text{sub}} = L/S + M$. The overall rate of the inner code is reduced by the factor $R_{\text{term}} = L/(L + SM)$, and thus, the available E_s/N_0 on the channel is reduced correspondingly.

Fig. 3 shows a typical situation where parts of the frame have been decoded by the sequential decoders. A control module keeps track of how many decoded symbols in each of the I RS words are currently available from the inner decoders. When enough symbols are available for decoding of a particular RS word, these symbols are copied to the corresponding RS decoder, which initiates a decoding attempt of that word. If the decoding is successful, the result is fed back by the control module to each of the inner decoders. From now on, the sequential decoders are forced to only follow paths in agreement with the branches pinpointed by this *side information* provided by the RS decoders. Because of the interleaved frame structure, each successfully decoded RS word will provide a *lighthouse* of J known bits for each $I \cdot J$ tree levels. As more and more symbols become available from the sequential decoders, some of the weaker RS words can be decoded and the results can be made available as more side information to the sequential decoders. In this way, the inner and outer decoders will effectively help each other in the decoding process toward a successful decoding of the full frame. Since the inner and outer decoders are working in parallel, outer decoding is constantly attempted. Thus, computational overflow can only occur if the number of errors made by the inner decoder plus the number of undecoded symbols exceeds the error correction capability of the weakest RS codewords. The freedom of exploring the code tree is eliminated when the decoder is in the forced sequence mode. Hence, it is crucial that each RS decoding is correct, and thus that T_i and $T_{i,\min}$ are not selected too small. If T_i is small, RS decoding might be incorrect, and thus the sequential decoder is constantly knocked off the correct path.

If, at some point in the decoding process, three consecutive RS words (when $J = 8$ and $M = 23$) have been decoded, then the state of the inner encoder is known repeatedly with intervals of $J \cdot I$ bits throughout the frame. Thus, the full frame is effectively split into $L/(J \cdot I)$ sub–subframes, each of which can be decoded independently, both in the forward and backward directions. To utilize this requires either $L/(J \cdot I)$ separate decoders or that the same forward and backward decoder-set can be reused to decode each sub–subframe in turn.

III. Theoretical Analysis of the FSSD Scheme

In order to predict the probability that a successful decoding of the strongest RS word will occur after C computations, we need information on how many bits have been decoded by each of the inner decoders.

In this section, we will extend the classic Pareto exponent analysis of the computational behavior of sequential decoding to find this probability, and we will do this by initially focusing only on a single unidirectional decoder and extend the results to unidirectional decoding in S subframes. Finally, we shall generalize the results to cover bidirectional decoding as well.

Although most of the path that is currently the best one is probably correct, the latest part have a provisional decoding status, where some of the bits may be incorrect and thus, this part may not contribute positively to an early RS decoding. We therefore focus on the number of actually decoded bits, α, i.e., they belong to the part of the path with a final decoding status, which is not changed by a back-search operation performed by the sequential decoder. This number can be found from the Pareto distribution, and it will be a lower bound to the number of bits that are in fact correct. Thus, as an estimate to the probability that the current path has ℓ correct information bits, we can use the probability that decoding of the first ℓ branches requires a total of less than C computations:

$$P[\alpha \geq \ell \,|\, C] \approx P\left[\sum_{i=1}^{\ell} C_i \leq C\right] = 1 - P\left[\sum_{i=1}^{\ell} C_i \geq C\right]. \quad (8)$$

For two random variables, C_1 and C_2, with probability density functions $h_1(C_1)$ and $h_2(C_2)$, the probability density function (pdf) of the sum, $C = C_1 + C_2$, is given by the convolution

$$\gamma(C) = h_1 * h_2 = \int_{-\infty}^{\infty} h_1(x)h_2(C-x)\,dx \qquad (9)$$

provided that C_1 and C_2 are independent. It is clear, however, that the random variables C_i are certainly *not* independent, and the correlation between C_1 and C_2 is unknown. Nevertheless, we shall start by presenting results for independent Pareto distributed variables, and then later we shall argue that the dependency in our sequential decoding process does not change the results significantly.

The pdf for the Pareto distribution is

$$f(x) = \frac{d}{dx}(1 - Ax^{-\rho}) = \rho A x^{-\rho-1}, \qquad x \geq 0$$

and since C_1 and C_2 are assumed to be independent and identically distributed Pareto random variables, we get for $\ell = 2$, the pdf

$$f_2(C) = \int_0^C f(x)f(C-x)\,dx \approx 2A\rho C^{-\rho-1}K. \qquad (10)$$

A numerical evaluation of the constant K reveals a value $K \to 1^-$ for $C \to \infty$. Unfortunately, the derivation of $f_\ell(C)$ for $\ell > 2$ becomes considerably more complicated, and no general solution for the sum of ℓ independent and identically distributed Pareto random variables has been derived. Only an asymptotic expression has been conjectured, and we have,

Conjecture 1—Sussman [12]: The sum of ℓ independent and identically distributed Pareto random variables with exponent ρ and parameter A, has a pdf $f_\ell(C)$ satisfying

$$\lim_{C\to\infty}(f_\ell(C) - \ell A\rho C^{-\rho-1}) = 0 \qquad (11)$$

and a cumulative distribution function (cdf) $F_\ell(C)$ satisfying

$$\lim_{C\to\infty}(F_\ell(C) - \ell A C^{-\rho}) = 0. \qquad (12)$$

Notice here, that the cdf defined by $F_\ell(C)$ is the *complement* of the usual definition of a cdf.

To support the conjecture we drew independent Pareto distributed random numbers and formed the sums $C = C_1 + C_2 + \cdots + C_\ell$ for different values of ℓ. We found, even for moderate to small values of C, a very close agreement between the resulting distribution of $P[(C_1 + C_2 + \cdots + C_\ell) > C]$ and the conjecture. We conclude that for independent C_i, Sussman's conjecture holds, even for moderate values of C.

Thus, assuming that the conjecture is also true for finite C, we find that the probability that ℓ bits have been decoded is $1 - F_\ell(C)$ where

$$F_\ell(C) = \begin{cases} 1, & C < (\ell A)^{1/\rho} \\ \ell A C^{-\rho}, & C \geq (\ell A)^{1/\rho}. \end{cases} \qquad (13)$$

We have also assumed, as already mentioned, that the C_i's are independent, although this is not a true description of the situation in hand. Before we can use the conjecture we therefore have to "confirm" or support the assumption that it is a good approximation to the actual distribution. Simulations for $P[(C_1 + C_2 + \cdots + C_\ell) > C]$, $\ell = 100, 200, \cdots, 2000$ and for $P[\alpha \geq \ell | C]$ with four fixed values of C were carried out with a Fano decoder for $E_s/N_0 = -4.95$ dB. These simulations strongly supported the linearity in ℓ shown in (13) and allow us to find empirical values of the constants ρ and A. Some values are shown in Table I.

TABLE I
PARETO EXPONENT, SLOPE, AND PARAMETER A FOR UNIDIRECTIONAL FANO DECODING

E_b/N_0	1.0 dB	1.2 dB	1.4 dB	1.6 dB
Calculated ρ	0.386	0.460	0.538	0.619
Measured ρ	0.48	0.53	0.58	0.66
Measured A	0.17	0.21	0.23	0.37

The only deviation was that the simulations reveal a larger value of $\rho = 0.66$, where the theoretical value is $\rho = 0.619$. This observation is explained by the fact that the Gallager function [see (5)] is derived for the full ensemble of codes. Since we have carefully selected the inner code, we get a better ρ than that predicted by (5). When the code is used for decoding in both directions, the *reversed* code, which has a slightly worse distance profile, will have a slightly worse ρ. Thus, the *average* ρ resulting from decoding in both directions will approximate the theoretical ρ.

Based on these observations, it seems fair to use a distribution of the form just derived as an approximation to the real distribution. We conclude that any discrepancy between the real and approximated distributions is accounted for by the constant A, not by the exponent ρ.

We now introduce the notation $g_i(\ell)$, which denotes the pdf for the sum of i independent and identically distributed [according to (13)] random variables, and $G_i(\ell)$ is the corresponding cdf. We then have

$$P[\alpha \geq \ell \,|\, C] = 1 - G_1(\ell) = 1 - F_\ell(C). \qquad (14)$$

From (14) we see that for a fixed number of computations C, the number α of correctly decoded bits is uniformly distributed on the interval $[0, C^\rho/A]$.

The uniform distribution described by (14) only takes values different from zero in the interval $[0, C^\rho/A]$, but ℓ can of course not be larger than the length of the (sub)frame, and we must therefore bound the distribution to take values only in $[0, L_{\text{sub}}]$. The upper bounding on ℓ will not affect how the decoder operates when the current decoder depth is far from the end of the subframe, but when the decoder touches the end and is in the zero state the current path is released and decoding is finished. This means that the last information bits of the subframe are not "decoded" under the same distribution as the first bits, but by avoiding *extremely* short subframes we may neglect this effect. The cdf will, under this assumption, increase linearly in the same way as $F_\ell(C)$ up to $\ell = L_{\text{sub}}$, where the probability $P[\alpha \leq \ell \,|\, C]$ will abruptly jump to one, while the real distribution may have a smoother change to one over the last few bits of the subframe. These observations may

be summarized for $C^\rho/A > L_{\text{sub}}$ by defining

$$\overline{G}_1(\ell) = \begin{cases} 0, & \ell < 0 \\ a\ell, & \ell \in [0,\, L_{\text{sub}}[\\ 1, & \ell \geq L_{\text{sub}} \end{cases} \tag{15}$$

where $a = AC^{-\rho}$. The corresponding pdf may be written

$$\overline{g}_1(\ell) = \begin{cases} a + (1 - aL_{\text{sub}})\delta(\ell - L_{\text{sub}}), & \ell \in [0,\, L_{\text{sub}}] \\ 0, & \text{otherwise} \end{cases} \tag{16}$$

where the delta function $\delta(\cdot)$ is introduced in order to ensure that the pdf integrates to one.

The expected value and variance of ℓ are given by

$$\overline{E}(\ell) = L_{\text{sub}} - aL_{\text{sub}}^2/2 \tag{17}$$

$$\overline{\text{Var}}(\ell) = (aL_{\text{sub}}^3/3) - (a^2L_{\text{sub}}^4/4). \tag{18}$$

We finally find

$$P[\alpha \geq \ell \,|\, C] = \begin{cases} 1 - G_1(\ell), & \ell \in \left[0,\, \dfrac{C^\rho}{A}\right], \quad \dfrac{C^\rho}{A} \leq L_{\text{sub}} \\ 1 - \overline{G}_1(\ell), & \ell \in [0,\, L_{\text{sub}}], \quad \dfrac{C^\rho}{A} > L_{\text{sub}} \end{cases} \tag{19}$$

as an approximation to the probability that a single decoder has decoded at least ℓ bits after C computations.

The distribution for the total number of bits decoded by S unidirectional decoders can be derived as follows using moment generating functions. The moment generating function for a random variable X uniformly distributed on $[0,\, b]$, where $b = a^{-1} = C^\rho/A$, is

$$\Phi_X(s) = E[\exp(-sX)] = (1 - \exp(-sb))/(sb).$$

This is simply the Laplace transform of the pdf of X. Let $Y = X_1 + X_2 + \cdots + X_j$ be a sum of independent identically distributed (i.i.d.) random variables. Then

$$\Phi_Y(s) = E[\exp(-s(X_1 + X_2 + \cdots + X_j)] = (\Phi_X(s))^j$$

where the latter equality follows from independence. The pdf for Y is recovered from the inverse Laplace transform, i.e.,

$$\begin{aligned} g_Y(\ell) &= \mathcal{L}^{-1}\Phi_Y(s) \\ &= \mathcal{L}^{-1}[(1 - \exp(-bs))^j/(bs)^j] \\ &= \mathcal{L}^{-1}\left[\frac{1}{(bs)^j}\sum_{i=0}^{j}\binom{j}{i}(-1)^i\exp(-ibs)\right] \\ &= b^{-j}\sum_{i=0}^{j}\binom{j}{i}(-1)^i\mathcal{L}^{-1}\frac{\exp(-ibs)}{s^j}. \end{aligned}$$

From standard Laplace transform tables we recognize

$$\mathcal{L}^{-1}\frac{\exp(-ibs)}{s^j} = \frac{(\ell - ib)^{j-1}}{(j-1)!}\,U(\ell - ib)$$

where $U(\cdot)$ is the unit step function. We now have the following lemma.

Lemma 1: The sum of j independent and identically distributed random variables with uniform pdf over $[0,\, C^\rho/A]$ has the pdf

$$g_j(\ell) = \frac{(AC^{-\rho})^j}{(j-1)!}\sum_{i=0}^{j}(-1)^i\binom{j}{i}\left(\ell - i\frac{C^\rho}{A}\right)^{j-1} \cdot U\left(\ell - i\frac{C^\rho}{A}\right) \tag{20}$$

and the cdf

$$G_j(\ell) = \frac{(AC^{-\rho})^j}{j!}\sum_{i=0}^{j}(-1)^i\binom{j}{i}\left(\ell - i\frac{C^\rho}{A}\right)^{j} \cdot U\left(\ell - i\frac{C^\rho}{A}\right). \tag{21}$$

Here, $G_j(\ell)$ follows by integration with respect to ℓ.

The case where we upper bound ℓ by the length of the subframe may be found in a similar way. In this case, we consider a random variable X with pdf $a + (1 - aL_{\text{sub}})\delta(x - L_{\text{sub}})$ for $x \in [0, L_{\text{sub}}]$. X then has moment generating function

$$\Phi_X(s) = \frac{a}{s}(1 - e^{-sL_{\text{sub}}}) + (1 - aL_{\text{sub}})e^{-sL_{\text{sub}}}.$$

The pdf for the sum of j i.i.d. versions of X can be recovered as the inverse Laplace transform of $\Phi_X(s)^j$, and we state the next lemma without proof.

Lemma 2: The sum of j independent and identically distributed random variables with pdf (16) and cdf (15) has the pdf

$$\overline{g}_j(\ell) = \sum_{i=0}^{j-1}\binom{j}{i}(1 - L_{\text{sub}}AC^{-\rho})^i \cdot g'_{j-i}(\ell) + (1 - L_{\text{sub}}AC^{-\rho})^j\delta(\ell - jL_{\text{sub}}) \tag{22}$$

and the cdf

$$\overline{G}_j(\ell) = \sum_{i=0}^{j-1}\binom{j}{i}(1 - L_{\text{sub}}AC^{-\rho})^i \cdot G'_{j-i}(\ell) + (1 - L_{\text{sub}}AC^{-\rho})^j U(\ell - jL_{\text{sub}}) \tag{23}$$

where

$$g'_{j-i}(\ell) = \frac{(AC^{-\rho})^{j-i}}{(j-i-1)!} \times \sum_{k=0}^{j-i}\left[(-1)^k\binom{j-i}{k}(\ell - (i+k)L_{\text{sub}})^{j-i-1} \times U(\ell - (i+k)L_{\text{sub}})\right] \tag{24}$$

and

$$G'_{j-i}(\ell) = \frac{(AC^{-\rho})^{j-i}}{(j-i)!} \times \sum_{k=0}^{j-i}\left[(-1)^k\binom{j-i}{k}(\ell - (i+k)L_{\text{sub}})^{j-i} \times U(\ell - (i+k)L_{\text{sub}})\right]. \tag{25}$$

We now arrive at the following lemma.

Lemma 3: The probability that unidirectional decoding in S subframes has reached a total depth of ℓ bits can be approximated by

$$P\left[\sum_{i=1}^{S}\mathcal{L}_i > \ell \,|\, C\right] \approx \begin{cases} 1 - G_S(\ell), & \dfrac{C^\rho}{A} \le L_{\text{sub}} \\ 1 - \overline{G}_S(\ell), & \dfrac{C^\rho}{A} > L_{\text{sub}} \end{cases}. \quad (26)$$

It is important to stress here that C denotes the number of computations performed by *each* decoder in the system. Thus, C becomes a measure of the actual decoding delay. If we perform unidirectional decoding in S subframes, the actual allocation of resources may be expressed by $C_{\text{total}} = S \cdot C$ computations. Thus, we fix the total resource allocation at C_{total} computations and distribute $C = C_{\text{total}}/S$ computations equally to each of the S decoders, even though some potential "operations" might be wasted when some of the subframes are finished. Whether or not these resources can be redirected to other subframes is a matter of the physical design of the decoder. It is also important to adjust the values of ρ and A for the different values of S to account for the additional losses due to terminating each subframe.

We will now turn our analysis to the bidirectional case where we will assume that the forward and backward decoders both perform equally well so that we can use the same values of ρ and A for the computational distribution in both directions. If the code does not comply with that assumption because the distance profile in one direction is somewhat better than in the other direction we may use the average values of ρ and A. Generalization to different values of ρ and A for the forward and backward decoders is straight-forward but tends to become a little messy, and thus, to make the derivation easier we have chosen to disregard this difficulty.

Let the random variables $\mathcal{F}$ and $\mathcal{B}$ denote the number of bits decoded by the forward and backward decoders, respectively. Since the forward and the backward decoders are independent and each simultaneously perform the same number of computations, C, the two random variables $\mathcal{F}$ and $\mathcal{B}$ are independent and identically distributed. When $\mathcal{F}+\mathcal{B}$ reaches L_{sub}, decoding of the subframe is finished, and the sum is therefore constrained to the interval, $(\mathcal{F}+\mathcal{B}) \in [0, L_{\text{sub}}]$.

When the forward and backward decoders meet and are in the same state, the two paths are merged and released, and decoding is finished. This affects the distribution of the C_i's in an interval around the merging point, but as we have already argued, this effect can almost be neglected when the length of the subframe is not too small.

Hence, the distribution function for $\mathcal{F}+\mathcal{B}$ is in principle found using the same methods we used for two unidirectional decoders in separate subframes. We can thus find the distribution of $\mathcal{F}+\mathcal{B}$ through three special cases, depending on the value of C, i.e., whether we have $2(C^\rho/A) < L_{\text{sub}}$ or $C^\rho/A < L_{\text{sub}} < 2(C^\rho/A)$ or $C^\rho/A > L_{\text{sub}}$.

The probability that decoding has been finished in the subframe after C_{total} computations is equivalent to the probability that the forward and backward decoders of the particular subframe have decoded a total of $\mathcal{F}+\mathcal{B}$ bits at least equal to the length of the subframe:

$$\text{Prob}\{\text{subframe finished}\} = P[\mathcal{F}+\mathcal{B} \ge L_{\text{sub}} \,|\, C] \quad (27)$$

which can easily be found from the above three special cases. Here it is again important that we use $C = C_{\text{total}}/2$ when calculating this probability.

For a fixed C, $P[\mathcal{F}+\mathcal{B} \ge L_{\text{sub}} \,|\, C]$ will result in an expression that indicates a doubling of the exponent in the probability of successful decoding of an entire subframe after C computations in *each* direction, compared to C computations in one unidirectional decoder. However, the total number of computations, $C_{\text{total}} = 2C$, is also doubled. Taking this into account together with the fact that these expressions are concerned with the number of decoded bits, the above should not lead one to the wrong conclusion that bidirectional decoding is equivalent to a doubling of the Pareto exponent in the usual sense, that is, the exponent for the distribution of the average number of computations of sequential decoding. This exponent is almost unaffected by the fact that we do bidirectional decoding.

It turns out that generalizing $P[\mathcal{F}+\mathcal{B} \ge \ell \,|\, C]$ to cover bidirectional decoding in S subframes is cumbersome. As an alternative to finding the exact expression we may instead make a small modification to how we interpret bidirectional decoding. If we assume that on the average the forward and backward decoders decode equally many bits in the same number of computations, we may thus actually assume two unidirectional decoders, each of length $L'_{\text{sub}} = L_{\text{sub}}/2$, instead of one bidirectional decoder of length L_{sub}. Equation (26) then applies directly, but we must use a subframe length of only $L'_{\text{sub}} = L_{\text{sub}}/2$ when evaluating it. The probability that a subframe has been decoded is then

$$P\{\text{subframe finished}\} = P[\mathcal{F}+\mathcal{B} \ge 2L'_{\text{sub}} \,|\, C] \quad (28)$$

where $C = C_{\text{total}}/2$. By using Lemma 3, we arrive at an approximate distribution function for bidirectional decoding

$$P\left[\sum_{i=1}^{S}(\mathcal{F}_i+\mathcal{B}_i) > \ell \,|\, C\right] \approx P\left[\sum_{i=1}^{2S}\mathcal{L}_i > \ell \,|\, C\right] \quad (29)$$

where we shall use $C = C_{\text{total}}/2S$. To compensate for the greater loss due to terminating $2S$ instead of S subframes one must also remember to adjust the constant A and the Pareto exponent ρ to have the values corresponding to bidirectional decoding in S subframes. Henceforth, we will use (29) as the distribution for bidirectional decoding in S subframes.

IV. Probability of Decoding the Strongest RS Word

Due to the terminating strings of zeros at the end of each subframe, the effective E_s/N_0 on the channel depends on the actual number of subframes that divide the full frame. As S is increased, the SNR seen from the inner decoders decreases correspondingly, making it harder to decode each bit, thus increasing C_{AV} and P_{OF}. However, another effect of increasing S is that the length of each subframe becomes shorter, and this minimizes the probability of more than one error burst in each subframe, and thus minimizes the

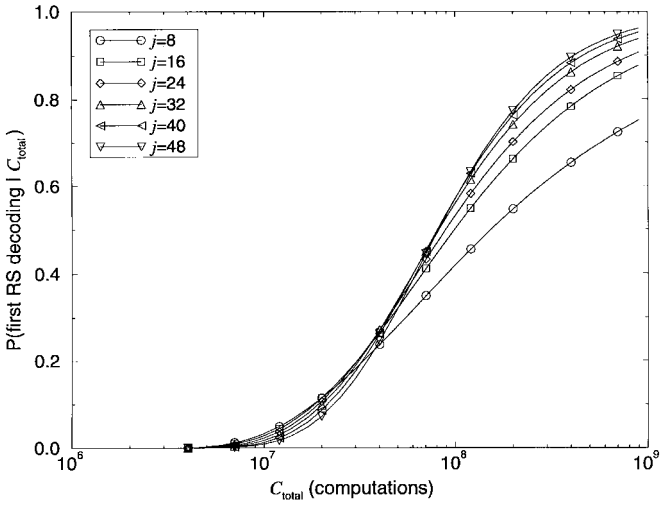

Fig. 4. Probability of having decoded $\ell = 15\,296$ bits after C_{total} computations (uniform RS profile). $E_b/N_0 = 1.0$ dB.

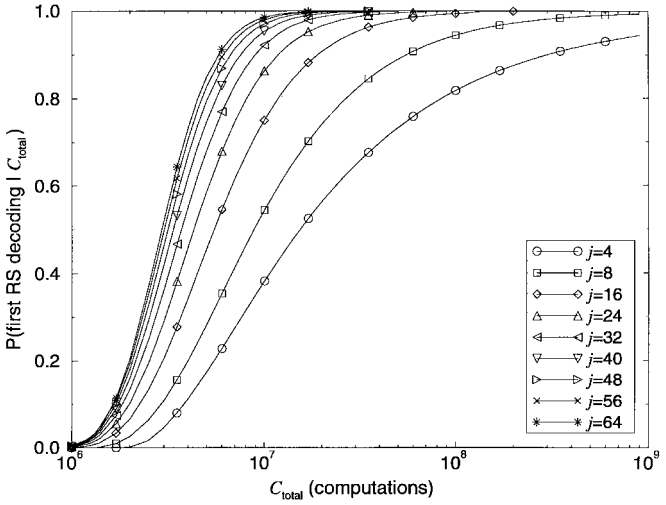

Fig. 5. Probability of having decoded $\ell = 13\,248$ bits after C_{total} computations (nonuniform RS profile). $E_b/N_0 = 1.0$ dB.

probability that the forward and backward decoders become stuck in two different error bursts.

The first RS decoding will be successful when the present number of errors t, and erasures e, satisfy (1) for the strongest RS word. The minimum number of *correct* symbols that is needed corresponds to the situation where the maximum number of erasures is present. If we assume this case, the following conditions must be met for correct decoding of the strongest RS word:

Number of erasures: $2(T_i - T_{\min,i})$

Minimum number of correct symbols: $n - 2T_i + T_{\min,i}$.

The situation that requires most correct symbols before decoding is successful occurs when there are no erasures present in the strongest RS word:

Number of erasures: 0

Minimum number of correct symbols: $n - T_i$.

If we assume that the S forward and backward decoders must decode a total of $\ell = (n-T)\cdot I \cdot J$ bits in order to guarantee that the first RS decoding is successful, then we have a probability, P_{RS}, of correct RS decoding after C_{total} computations equal to the probability that the $2S$ decoders have decoded a total number of bits at least equal to ℓ. This probability is given by Lemma 3 and we have

$$P_{\text{RS}} = P\left[\sum_{i=1}^{2S} \mathcal{L}_i > \ell \,|\, C\right] \tag{30}$$

where the value of A may be found using Table I by interpolating between the values shown for the various E_s/N_0 depending on the actual number of subframes.

As an example we may consider profiles like

$$\begin{aligned} \text{RS}_1^8 = &[16,\ 16,\ 16,\ 16,\ 16,\ 16,\ 16,\ 16], \\ &[16,\ 16,\ 16,\ 16,\ 16,\ 16,\ 16,\ 16] \\ \text{RS}_2^8 = &[10,\ 48,\ 10,\ 6,\ 6,\ 6,\ 6,\ 6], \\ &[10,\ 16,\ 10,\ 6,\ 6,\ 6,\ 6,\ 6] \end{aligned}$$

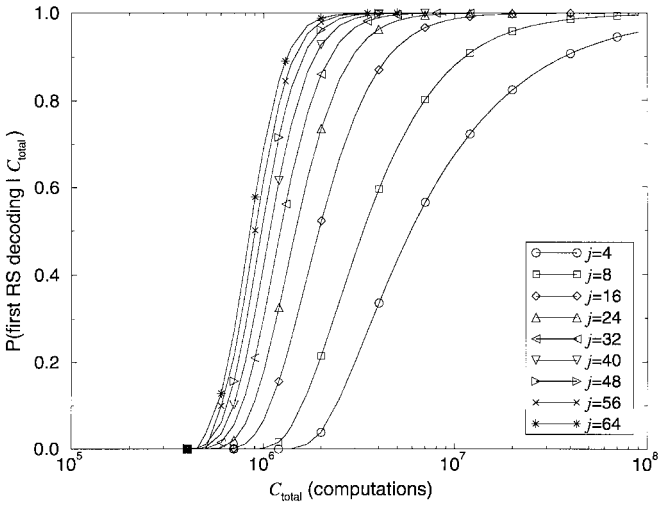

Fig. 6. Probability of having decoded $\ell = 11\,200$ bits after C_{total} computations (nonuniform RS profile). $E_b/N_0 = 1.0$ dB.

where in the first case we have a standard uniform profile with $T_i = 16$, $R_{\text{outer}}^1 = 0.875$ and in the second case we have a nonuniform profile with the strongest code having $T_2 = 48$ and $R_{\text{outer}}^2 = 0.904$.

We will first assume the worst-case situation where on the average we need $n - T_i$ correct symbols for the strongest RS code, totaling $(n - T_i) \cdot J \cdot I$ correct bits in the full frame. This is also the total number of bits that must be decoded by the S forward and backward decoders. By letting j denote the number of inner decoders we have $j = S$ for unidirectional decoding and $j = 2S$ for bidirectional decoding. When evaluating (30) for up to $j = 64$ inner (unidirectional) decoders and the uniform or nonuniform outer code profile, we obtain the plots shown in Figs. 4 and 5 for $E_b/N_0 = 1.0$ dB. Next, we assume the optimum case where on the average we need $n - 2T_i + T_{\min,i}$ correct symbols in each RS word, which for the nonuniform profile totals 11 200 correct bits in the full frame. The results for the nonuniform profile are shown in Fig. 6.

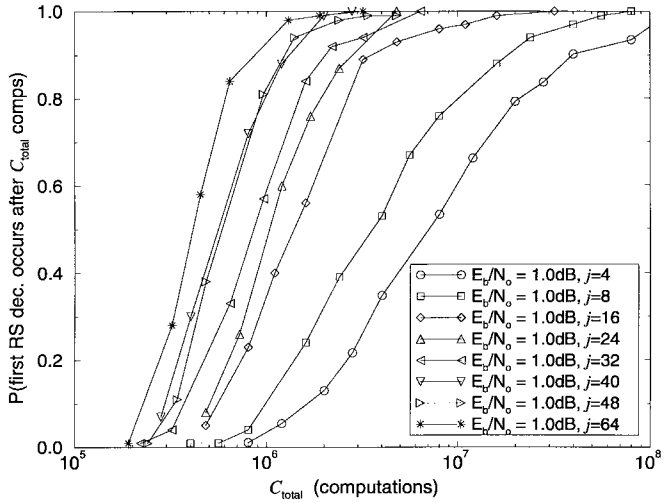

Fig. 7. Probability that the first RS decoding had occurred after C_{total} computations. Nonuniform RS profile. Forward decoding only. Simulation results for $E_b/N_0 = 1.0$ dB.

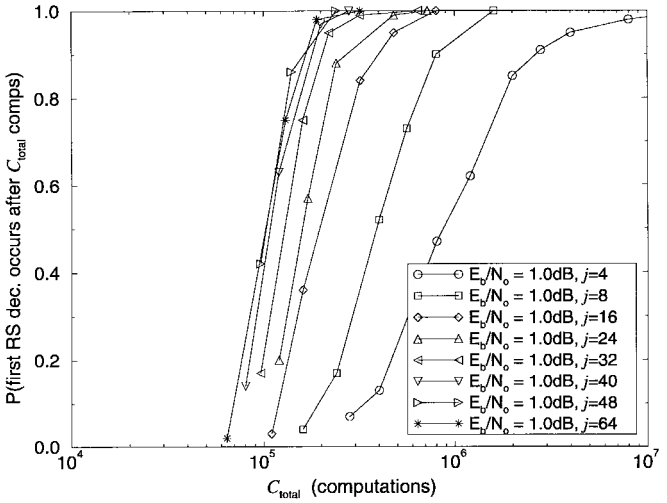

Fig. 8. Probability that the first RS decoding had occurred after C_{total} computations. Nonuniform RS profile. Forward and backward decoding. Simulation results for $E_b/N_0 = 1.0$ dB.

In all cases, we see that the more subframes we use, the more bits are decoded in the same number of computations. Consequently, the first RS decoding becomes more likely to occur early when we use a large number of subframes. However, due to the increased loss from the terminating strings, the improvement becomes smaller for each additional increment of j, and for $j > 64$ the extra improvement is negligible.

For the uniform case, we need at least 10 times as many computations to get the same probability that RS decoding will occur than for the nonuniform case. This is caused by the fact that for the uniform profile we need a significantly larger number of correct bits decoded by the inner decoders. In the FSSD scheme this is a waste of resources since forced sequence decoding is thus initiated relatively late in the process, and probably too late to become really useful. The nonuniform profile thus is a much better choice for the FSSD scheme.

From the calculations we conclude that it is not feasible to go beyond $j = 64$ decoders since the extra gain is negligible. From a practical point of view, an even smaller number might be better, since the complexity of building a hardware decoder with many inner decoders is considerable. Thus one might be better off by choosing a value of $j = 48$ or lower, which corresponds to $S = 24$ bidirectional decoders.

To support the theory we performed a number of simulations with the nonuniform profile. To obtain results that comply as closely as possible with the theory, we first simulated a system with $2S$ unidirectional decoders instead of S bidirectional decoders. The results are shown in Fig. 7. Since the RS decoders perform errors and erasures decoding we actually have a situation close to the "optimum case" described above, i.e., corresponding to the results shown in Fig. 6. To account for the extra terminating strings introduced by assuming $2S$ unidirectional decoders instead of S bidirectional decoders we have adjusted A and ρ to match the bidirectional case. The reason for the slightly jagged curves seen in Fig. 7 may be ascribed to uncertainty in the simulations.

First, we notice that the form of the curves in Fig. 7 is almost identical to the form of the curves shown in Figs. 5 and 6, which support our theoretically derived expressions.

Further, we see the same tendency that more subframes gives better results, just as was indicated by the theory. We should still choose larger values of S in order to have the first RS codeword decoded as early as possible. The simulation results also show that the gain from continuously increasing S becomes smaller, but the effect is not quite as clear as was observed theoretically.

There is a small discrepancy when we have many subframes. The simulation results show that we actually need fewer computations than predicted by the theory. One reason for this discrepancy may be that each bit along the current paths of the $2S$ decoders does not need to be completely *decoded*. The depths of the decoders actually provide more correct bits than we predict from the theory, and this has the effect that correct RS decoding can take place sooner than we may expect from (30).

We have assumed that bidirectional decoding in S subframes may be well approximated by $2S$ unidirectional decoders. To see how well this assumption fits we have performed simulations of a true bidirectionally decoded system. The form of the resulting curves shown in Fig. 8, is again very similar to the theoretically derived ones shown in Figs. 5 and 6, and we see this as strong support for our assumption. Although the curves are roughly as expected, we see that the bidirectional decoders obtain the first RS decoding in a smaller number of operations than twice as many unidirectional decoders.

The main reason for this is that all the decoders may advance further than $L_{\text{sub}}/2$ bits, and thus the "waste" of operations that takes place for $2S$ unidirectional decoders when some of them have reached the subframe length is significantly reduced. For a particular subframe, a single burst which is difficult to decode is approached from both sides when we use bidirectional decoding. In the unidirectional decoder case the noise burst will only be approached from one side, and thus the part of the subframe that lies beyond the burst is left

undecoded for the period of time it takes the sequential decoder to penetrate the burst, which may also delay the first RS decoding. The overall result is that the bidirectional decoders provide more correct bits in the same number of computations compared to the unidirectional case, and thus also obtain an earlier first RS decoding, as is seen by comparing Figs. 7 and 8. As the SNR worsens, the number of noise bursts increases correspondingly and the benefits from doing bidirectional decoding becomes more and more prominent.

For both the unidirectional and the bidirectional decoding cases, it is evident that by choosing an even stronger RS word one can obtain better results than those shown in the previous figures. What is also obvious though, is that this extra power must be taken from the remaining words of the profile, or else it will reduce the overall rate of the outer code, which might be undesirable.

Currently, the behavior of the FSSD scheme in the period between the first successful RS decoding and the time when the frame is fully decoded or has become an erasure, is not well known.

In subsequent simulations we have observed that there is a relatively clear optimum of $S = 15$ subframes when we perform bidirectional decoding in several subframes at very low SNR. For $S > 20$, we start seeing more and more overflows even at SNR's of about 1.0 dB. This means that the loss due to the terminating strings has a not well understood influence on the behavior of the scheme. This observation is somewhat in contrast to the results obtained above. It is clear, however, that the other RS words in the profile play an important role with respect to how the system behaves between the first and last successful RS decoding. One way to avoid this problem might be to divert more error and/or erasure correcting power to the first and third RS words of the outer profile. This will have the effect that these words will also quickly be decoded such that inner sub–subframe decoding can be initiated as early as possible. Depending on how much we are willing to compromise on the overall rate of the outer codes, the error and erasure correcting power of the remaining five RS words must be adjusted to comply with that change.

In Fig. 9, we have shown simulation results of C_{AV} for the full FSSD scheme, i.e., at the time when the iterative process between the inner and outer decoders terminates, either because the frame has been completely decoded or because of computational overflow. The results include a few sparse profiles, and these are introduced to reduce the number of different RS words in the profile, and thus also to reduce complexity. In all cases we used $S = 15$ subframes, and simulated a total of about $1.4 \cdot 10^6$ information bits. The observed fraction of frames that suffered an overflow, P_{OF}, is indicated at each data point. No indication means that no overflows were observed.

For the nonuniform profiles we notice that even at $E_b/N_0 =$ 0.8 dB the average number of computations needed to decode one bit, is more than 30 times less than the complexity of a one-pass Viterbi decoded Galileo system [13] which performs 16 384 computations per decoded bit. In fact, only one frame needed more computations per bit (19 213) than the Viterbi decoded Galileo system. The uniform profile performs

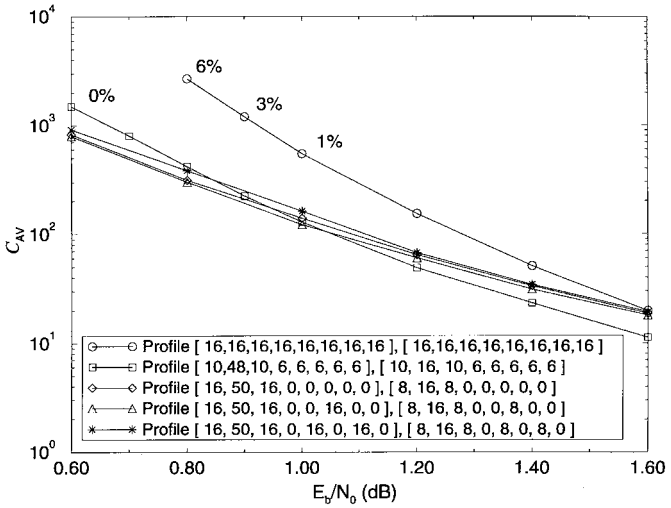

Fig. 9. Average number of computations per decoded bit for different profiles.

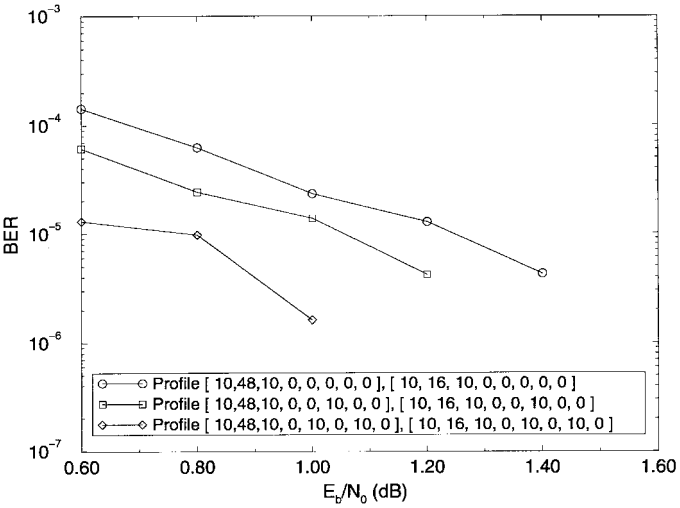

Fig. 10. BER for different sparse profiles.

considerably worse, as we expect from the theoretical results seen above.

Also, for the nonsparse profiles, no decoding errors at all were observed in any of the frames that did not suffer an overflow. For the sparse profiles, however, the decoding errors from the inner decoders slip through the holes in the profile. Fig. 10 shows the BER after outer decoding for the sparse profiles, and as may be seen, the BER increases with increasing sparseness of the profile.

These observations indicates that for the FSSD scheme employing nonsparse profiles, BER is not the parameter by which the performance should be measured. The important parameter is the probability of overflow. When a sparse profile is used C_{AV} can for the lowest SNR's be further reduced compared to the nonsparse case, at the cost of a nonzero BER.

We see that both low C_{AV} and almost neglectable P_{OF} can be obtained for all $E_b/N_0 \geq 0.6$ dB. Furthermore, depending on the sparseness of the profile, zero or very low BER can also be obtained.

A final comment on the practical implementation may be in order. Parallel decoding is assumed, in which outer decoders are constantly attempting decoding from partial decisions made by inner decoders. It is implicit in the analysis and simulations that the outer decoders react immediately, but this is not possible in a practical scheme where the outer decoder needs time to decode. During that lag in time the inner decoders continue decoding and they may have corrected enough symbols for the strongest outer decoder to take advantage before the outer decoder is ready to start a new attempt. Thus, the actual number of computations for a practical scheme may be slightly larger, depending, of course, on how fast the outer decoders operate.

V. Conclusion

The average load, or complexity, of the FSSD system can be considered to be very low compared to other systems with similar performance, e.g., the Galileo system, [13] and the Turbo system, [14]. The Galileo system has an overall rate similar to our system and similar performance, while the Turbo scheme can achieve better performance with a higher overall rate. Based on the method used in [14] where complexity is measured as a rough count of the number of additions and multiplications needed to decode one bit, we estimate the complexity of the proposed FSSD scheme to be much lower, on the average, compared to the latter two systems.

The overflow probability of the FSSD system can to some extent be *selected* by the user simply by allowing a larger upper limit on the amount of available resources. Since overflows only occur occasionally, with increasing intensity as SNR drops, a sufficiently large buffer for storing incoming frames during decoding of potential overflow frames, in combination with a large computational limit, can almost eliminate overflows. Another approach would be to temporarily store overflowed frames for later off-line decoding. In both cases the *average* load per subframe would still be acceptable. Considering these arguments, the FSSD scheme appears to be a good choice in many applications that are not time critical.

One major advantage of the FSSD scheme is its high degree of flexibility and adaptability. In high SNR applications it will produce decoding results with very short average delay, and due to the small load it can operate at a high data rate. In low SNR applications, we normally have a somewhat lower data rate, and the decoder may thus adapt to the situation and still produce reliable output, but with a somewhat larger average delay. In deep space applications, such as a mission to multiple planets, the amount of down-loaded information can be adjusted to accommodate the available SNR at any given time, and our system can automatically adapt to the changing conditions as more distant planets are visited. This is not possible for a Viterbi decoder which always uses the same number of computations, regardless of the SNR in hand, and it must therefore be designed for the worst case situation.

It is worth noting that these results are obtained well above the cutoff rate that for many years was believed to be the *practical* limit of sequential decoding. With the FSSD scheme we operate about 1.7 dB below the SNR that marks the cutoff rate for the convolutional code alone, and about 1.25 dB below the cutoff rate for the full system. The key that makes these results possible is the iterative process where the outer decoders provide side information to aid continued decoding, in combination with a large number of individually and bidirectionally decoded subframes in the inner system.

Although the performance of the system presented in this paper compares favorably to most other systems, we believe that even better results may be obtained.

One issue that needs to be addressed is the rate allocation of the outer code. Here, we have only investigated the possibility of using one strong RS code, such that the rate of the outer code is kept approximately equal to the rate of the RS code profile used in the Galileo system, allowing for easier comparisons. However, by introducing a second (or several) strong outer codes we might be able to improve the overall performance of the FSSD system on the cost of a small reduction in the overall rate. This rate reduction may on the other hand be counter affected by increasing the interleaving degree. In combination with a large number of subframes more strong RS codes improve the probability that the iterative process is initiated early, which is exactly what is needed to get good performance with the FSSD scheme.

To avoid the relatively few decoder errors made by the sequential decoders when we use a sparse profile, we might use a longer constraint length convolutional code. By evaluating the expressions of Section IV for a memory $M = 40$ code, e.g., we see indications that there might be an optimum number of subframes with respect to when the first successful RS decoding occurs. This value is around $S = 16$. A new search for longer nonsystematic codes and codes with lower rates with good properties (a large d_f and a good DP in both directions) is thus needed.

Theoretically, the behavior of the scheme beyond the first RS decoding is still not well understood and deserves further investigation before the scheme is fully documented.

Acknowledgment

The authors would like to thank two anonymous reviewers for their thorough reviews, and for the excellent suggestions on how to improve the paper.

References

[1] Consultative Committee for Space Data Systems, "Recommendation for space data system standards, telemetry channel coding," CCSDS 101.0-B-3, Blue Book, May 1992.

[2] O. M. Collins and M. Hinzla, "Determinate-state convolutional codes," *IEEE Trans. Commun.*, vol. 41, pp. 1785–1794, Dec. 1993.

[3] E. Paaske, "Improved decoding for a concatenated coding system recommended by CCSDS," *IEEE Trans. Commun.*, vol. 38, pp. 1138–1144, Aug. 1990.

[4] F. Jelinek and J. Cocke, "Bootstrap hybrid decoding for symmetrical binary input channels," *Inform. Contr.*, vol. 18, no. 3, pp. 261–298, Apr. 1971.

[5] R. J. McEliece and L. Swanson, "On the decoder error probability for Reed–Solomon codes," *IEEE Trans. Inform. Theory*, vol. IT-32, pp. 701–703, 1986.

[6] R. Johannesson, "Some rate 1/3 and 1/4 binary convolutional codes with an optimum distance profile," *IEEE Trans. Inform. Theory*, vol. IT-23, pp. 281–283, Mar. 1977.

[7] J. J. Bussgang, "Some properties of binary convolutional code generators," *IEEE Trans. Inform. Theory*, vol. IT-11, pp. 90–100, Jan. 1965.
[8] G. D. Forney, "Convolutional codes I: Algebraic structure," *IEEE Trans. Inform. Theory*, vol. IT-16, pp. 720–738, Nov. 1970.
[9] K. Zigangirov, "Some sequential decoding procedures," *Prob. Peredach. Informatsii*, vol. 2, pp. 13–25, 1966.
[10] F. Jelinek, "A fast sequential decoding algorithm using a stack," *IBM J. Res. Develop.*, vol. 13, pp. 675–685, Nov. 1969.
[11] R. M. Fano, "A heuristic discussion of probabilistic decoding," *IEEE Trans. Inform. Theory*, vol. IT-9, pp. 64–74, Apr. 1963.
[12] S. M. Sussman, "Analysis of the Pareto model for error statistics on telephone circuits," *IEEE Trans. Commun. Technol.*, vol. COM-11, pp. 213–221, June 1963.
[13] S. Dolinar and M. Belongie, "Enhanced decoding for the Galileo low-gain antenna mission," in *Proc. 1994 IEEE Int. Symp. Information Theory*, Trondheim, Norway, June/July 1994, p. 344.
[14] J. D. Andersen, "The TURBO coding scheme," Rep. IT-146, Inst. of Telecommun., Technical Univ. of Denmark, DK-2800 Lyngby, June 1994, revised Dec. 1994.

Ole Riis Jensen was born in Denmark on Aug. 9, 1968. He received the M.Sc. and Ph.D. degrees in electrical engineering from the Technical University of Denmark, Lyngby, in 1992 and 1997, respectively.

Since 1997, he has been with the Technical University of Denmark, where he is a Research Professor at the Department of Telecommunication. He is now primarily working with multimedia problems, including synchronization of multimedia events, image, and video coding (JPEG-2000, MPEG-4), error resilience in video coding, and real-time streaming of video and audio over packetized networks.

Erik Paaske was born in Denmark on April 14, 1938. He received the M.Sc. and Ph.D. degrees in electrical engineering from the Technical University of Denmark, Lyngby, in 1963 and 1969, respectively.

From 1963 to 1965, he was employed by F. L. Smidth and Co. A/S working with factory automation. Since 1965, he has been with the Technical University of Denmark, where he is now an Associate Professor at the Department of Telecommunication, teaching and conducting research on information theory and digital communication. His main interest is error-correcting coding, including architectures and VLSI design for encoders and decoders.